

Anthony Vigil

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/16/2022 | Anthony Vigil | Design |

**Instructions**

Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

## [Executive Summary](#_sbfa50wo7nsh)

A new client, The Gaming Room, wants to develop a web-based game that serves multiple platforms based on their current game. Their current game is called “Draw It or Lose It” and is currently only available as an android app. The game consists of teams competing to guess what is drawn. It will have people from the teams go four rounds at a minute each. When a picture or image is pulled from the library of images, one team will have 30 seconds to guess. When a team does not guess within the time limit, the remaining teams will have the opportunity to make one guess with a 15 second time limit.

## [Design Constraints](#_2et92p0)

* Game requires one or more teams involved
* Multiple players assigned to each team
* Must have unique game and team names to allow users to check if the name is available
* Must only have one instance of the game exist in memory at any time.
* Be able to run on several platforms

The design constraints are also requirements to follow during the process of development. The constraints focus on the functionality of the game, but further observation is required for the process of application development. The Gaming Room requires the game to run on multiple platforms such as Apple, Windows, and Linux devices. The game is currently available as an android app. To complete this task, we must research and edit the current code to work with other programming languages; possibly with integration of one language to another.

## [System Architecture View](#_ilbxbyevv6b6)

Please note: There is nothing required here for these projects, but this section serves as a reminder that describing the system and subsystem architecture present in the application, including physical components or tiers, may be required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

## [Domain Model](#_8h2ehzxfam4o)

As shown in the diagram below, Entity is creating a relationship between the Game, Team and Player classes. The UML diagram shows there is inheritance; allowing each class to use the same references such as “id” or “name”. Game has the team list and GameService has games list. Both Team and Player are considered aggregation types under UML due to having an instance of one class but also referencing an instance of another class. The diagram shows GameService having a reference of Game, Game referencing Team, and Team referencing Player; from left to right. ProgramDriver and Singleton are not under the superclass Entity.

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [Evaluation](#_2o15spng8stw)

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s requirements and look at the situation holistically, as it all has to work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Configurable access, server, and changes due to its flexible terminal.  Characteristics: Highly used for web browsing  Advantages:  Many options for web hosting and upgradability.  Disadvantages:  Not preferred for web hosting | Configurable server, access, cost-efficient, and same flexibility of terminal commands as a Mac.  Characteristics:  Very secure  Advantages:  Preferred choice for web hosting services, security flaws are found easier.  Disadvantages:  Hard to find applications that support web hosting requirements. | Tons of software available when compared to other OS  Characteristics:  Closed platform, dominant when compared to other platforms.  Advantages:  Gives less loading time, more comfortable than others, high resource requirements.  Disadvantages:  Prone to viruses, not much support | Limitation when it comes to servers. It is preferred to have stationary server as it allows for easy tracking.  Characteristics:  Very portable, popular for on the-go.  Advantages:  Cost-effective, more range, and better compatibility.  Disadvantages:  Lacking security and poor connectivity on some mobile devices. |
| **Client Side** | Cost is more than Windows. Requires moderate expertise and time. | The least cost, minimum cost. Requires maximum expertise and time.  Has tons of open-source software. | Cost is moderate, similar to Mac. Requires minimum expertise and time.  Known to be a great platform for developing windows apps or websites. | Cost is average, requires maximum expertise and time. Difficult to set up. Updates are shown to both developers and clients.  Known mainly to be used on phones or tablets. UI is designed according to the device’s limitations such as screen size or controls. |
| **Development Tools** | The language commonly used in Mac OS is Objective-C; It is used for programming. | Linux supports many programming languages. Linux can work with Eclipse, Visual Studio and notepad++  Programming languages:  C, C++, Java, JavaScript, Vala, CSS, Python, Ruby, PHP, or HTML. | Microsoft utilizes C++, it is the most used for building applications. It is known to be the general programming language. Windows Visual Studio is an IDE that can be used to program in C++. | Mobile app development utilizes Swiftic.  The programming language used for Android app development is Java; Making it the most common language in android mobile devices. |

## Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

1. **Operating Platform**: I would recommend The Gaming Room to begin with Windows as it requires the minimum amount of expertise and time. Windows has many IDEs to utilize for your preferred programming language. Also, a variety of software is available when compared to other OS.
2. **Operating Systems Architectures**: Graphical User Interfaces (GUI) are shown by Windows-based applications when accessing resources in the system. The applications utilize web services, graphics, or multimedia. The services are accessed by a server or a user account.
3. **Storage Management**: Saving data and managing files is easier on Windows, as it allows the user to choose save locations for applications. Having its own storage system, it allows users to create files and safely store them in a location of preference. There is also a cloud storage that can be used to save files on the go.
4. **Memory Management**: Creating the game will require a library or database with many pictures. The pictures can easily be stored outside the default Windows picture folder by creating its own folder. This helps keep the project organized and allows easier functionality when using an IDE.
5. **Distributed Systems and Networks**: For cross-platform game creation, Develop 4 can be utilized to publish a game. It can be exported into the platform of your choice such as Android, iOS or the web. Testing the servers for player capacity and power outages is required.
6. **Security**: Windows comes with Windows Security Essentials, a security protection software. It protects the user data but is not recommended to use for security. Windows Security Essentials scans for viruses and malware but it is recommended to use another source for better security protection.